

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-007387

(43)Date of publication of application : 12.01.1999

(51)Int.Cl.

G06F 9/38  
G06F 9/30

(21)Application number : 09-159048

(71)Applicant : MATSUSHITA ELECTRIC IND CO  
LTD

(22)Date of filing : 16.06.1997

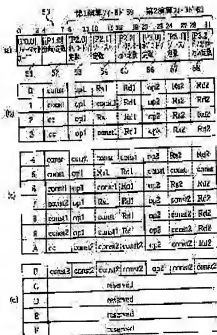
(72)Inventor : TAKAYAMA SHUICHI  
HIGAKI NOBUO

## (54) VLIW PROCESSOR

## (57)Abstract:

PROBLEM TO BE SOLVED: To provide a VLIW(very long instruction word) processor to execute an instruction structured with excellent code efficiency with comparatively short word and capable of simultaneously specifying many operations.

SOLUTION: Fields 52, 59, 60 to specify three pieces of operation at maximum to the instruction 50 of 32 bit length is provided. Only an operation code 'cc' to specify branch operation in which a stored value of an implicitly specified constant register is defined as a branching destination address or a constant 'const' to be set in the constant register is placed in a P 1.0 field 52. Which of them is specified by a formal code placed in a P 0.0 field 51 of four bit length.



(51) Int.Cl.<sup>6</sup> 識別記号  
G 0 6 F 9/38 3 1 0  
9/30 3 5 0

F I  
C 0 6 F 9/38 3 1 0 A  
3 1 0 H  
9/30 3 5 0 F

審査請求 未請求 請求項の数27 O L (全 18 頁)

(21) 出願番号 特願平9-159048

(22) 出願日 平成 9 年(1997) 6 月16 日

(71) 出願人 000003821

松下電器産業株式会社

大阪府門真市大字門真1006番地

(72) 発明者 高山 秀一

大阪府門真市大字門真1006番地 松下電器  
産業株式会社内

(72) 発明者 檜垣 信生

大阪府門真市大字門真1006番地 松下電器  
産業株式会社内

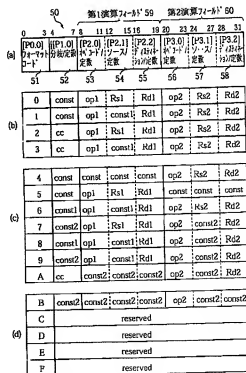
(74) 代理人 弁理士 中島 司朗

(54) 【発明の名称】 V L I W プロセッサ

(57) 【要約】

【課題】 比較短的な語長の命令であって、かつ、同時に多くのオペレーションを指定することができるコード効率のよい構造を有する命令を実行する V L I W プロセッサを提供する。

【解決手段】 32ビット長の命令50に最大3個のオペレーションを指定するフィールド52、59、60を設ける。P1、0フィールド52には、暗黙的に指定された定数レジスタ36の格納値を分岐先番地とする分岐オペレーションを指定するオペコード“cc”のみ、又は、定数レジスタ36にセットする定数“const”が置かれる。いずれが置かれているかは、4ビット長のP0、0フィールド51に置かれたフォーマットコードによって特定される。



## 【特許請求の範囲】

【請求項1】 複数のオペレーションフィールドからなる命令を実行するVLIWプロセッサにおいて、

前記オペレーションフィールドの大きさが不均一であり、かつ前記命令の命令語長は前記命令の持つオペレーションフィールドの数で割り切れないものであることを特徴とするVLIWプロセッサ。

【請求項2】 複数のオペレーションフィールドからなる命令を実行するVLIWプロセッサにおいて、前記オペレーションフィールドの大きさが不均一であり、かつ前記命令は3つのオペレーションフィールドを32ビットの命令語長中に持つものであることを特徴とするVLIWプロセッサ。

【請求項3】 複数のオペレーションフィールドからなる命令を実行するVLIWプロセッサにおいて、前記オペレーションフィールドのうち少なくとも1つはオペランドの数が異なるものであり、かつ前記命令の命令語長は前記命令の持つオペレーションフィールドの数で割り切れないものであることを特徴とするVLIWプロセッサ。

【請求項4】 複数のオペレーションフィールドからなる命令を実行するVLIWプロセッサにおいて、前記オペレーションフィールドのうち少なくとも1つはオペランドの数が異なるものであり、かつ前記命令は3つのオペレーションフィールドを32ビットの命令語長中に持つものであることを特徴とするVLIWプロセッサ。

【請求項5】 複数のオペレーションフィールドからなる命令を実行するVLIWプロセッサにおいて、前記オペレーションフィールドのうち1つはオペコードのみからなり、かつ前記命令の命令語長は前記命令の持つオペレーションフィールドの数で割り切れないものであることを特徴とするVLIWプロセッサ。

【請求項6】 複数のオペレーションフィールドからなる命令を実行するVLIWプロセッサにおいて、前記オペレーションフィールドのうち1つはオペコードのみからなり、かつ前記命令は3つのオペレーションフィールドを32ビットの命令語長中に持つものであることを特徴とするVLIWプロセッサ。

【請求項7】 2個以上のオペレーションフィールドを含む命令を解読し実行するVLIWプロセッサであって、

第1の前記オペレーションフィールドにはオペレーションの種類を指定する1個のオペコードのみが置かれ、

第2の前記オペレーションフィールドには1個のオペコードとオペレーションの対象となるデータを指定する1個以上のオペランドとの組が置かれ、

前記第1のオペレーションフィールドに置かれたオペコードを解読する第1の解読手段と、

前記第1の解読手段による解読結果に基づいて前記オペ

コードによって指定されたオペレーションを実行する第1の実行手段と、

前記第2のオペレーションフィールドに置かれたオペコードを解読する第2の解読手段と、

前記第2の解読手段による解読結果に基づいて前記オペランドによって指定されたデータに対して前記オペコードによって指定されたオペレーションを実行する第2の実行手段とを備えることを特徴とするVLIWプロセッサ。

【請求項8】 前記第1のオペレーションフィールドの桁数は前記第2のオペレーションフィールドの桁数よりも小さいことを特徴とする請求項7記載のVLIWプロセッサ。

【請求項9】 前記第1のオペレーションフィールドに置かれたオペコードの桁数は前記第2のオペレーションフィールドに置かれたオペコードの桁数と等しいことを特徴とする請求項8記載のVLIWプロセッサ。

【請求項10】 前記命令に含まれるオペレーションフィールドは3個であり、

第3の前記オペレーションフィールドは前記第2のオペレーションフィールドと同じ桁数であって1個のオペコードと1個以上のオペランドとの組が置かれ、

前記VLIWプロセッサはさらに、

前記第3のオペレーションフィールドにオペコードが置かれた場合に前記オペコードを解読する第3の解読手段と、

前記第3の解読手段による解読結果に基づいて前記オペランドによって指定されたデータに対して前記オペコードによって指定されたオペレーションを実行する第3の実行手段とを備えることを特徴とする請求項9記載のVLIWプロセッサ。

【請求項11】 前記第1の実行手段は、実行すべき命令の流れを制御することを特徴とする請求項10記載のVLIWプロセッサ。

【請求項12】 前記第2の実行手段は、前記第2のオペランドフィールドに置かれたオペランドによって指定されたデータの転送を制御し、

前記第3の実行手段は、前記第3のオペランドフィールドに置かれたオペランドによって指定されたデータの算術論理演算を実行することを特徴とする請求項11記載のVLIWプロセッサ。

【請求項13】 2個以上のオペレーションフィールドを含む命令を解読し実行するVLIWプロセッサであって、

第1の前記オペレーションフィールドにはオペレーションの種類を指定する1個のオペコードのみ又は定数のみが置かれ、

第2の前記オペレーションフィールドには1個のオペコードとオペレーションの対象となるデータを指定する1個以上のオペランドとの組又は定数のみが置かれ、

前記第1のオペレーションフィールドにオペコードが置かれた場合に前記オペコードを解読する第1の解読手段と、

前記第1の解読手段による解読結果に基づいて前記オペコードによって指定されたオペレーションを実行する第1の実行手段と、

前記第2のオペレーションフィールドにオペコードが置かれた場合に前記オペコードを解読する第2の解読手段と、

前記第2の解読手段による解読結果に基づいて前記オペランドによって指定されたデータに対して前記オペコードによって指定されたオペレーションを実行する第2の実行手段とを備えることを特徴とするVLIWプロセッサ。

【請求項14】 前記命令はさらに、前記第1及び第2のオペレーションフィールドそれぞれに定数のみが置かれているか否かを指定するフォーマットコードが置かれたフォーマットフィールドを含み、

前記VLIWプロセッサはさらに、前記フォーマットコードを解読するフォーマット解読手段と、

前記フォーマット解読手段により前記第1、第2及び第3の少なくとも1つのオペレーションフィールドに定数のみが置かれていると解読された場合に、その定数を取り出して記憶する定数記憶手段とを備えることを特徴とする請求項13記載のVLIWプロセッサ。

【請求項15】 前記第1のオペレーションフィールドの桁数は前記第2のオペレーションフィールドの桁数よりも小さいことを特徴とする請求項14記載のVLIWプロセッサ。

【請求項16】 前記第1のオペレーションフィールドに置かれたオペコードの桁数は前記第2のオペレーションフィールドに置かれたオペコードの桁数と等しいことを特徴とする請求項15記載のVLIWプロセッサ。

【請求項17】 前記命令に含まれるオペレーションフィールドは3個であり、

第3の前記オペレーションフィールドは前記第2のオペレーションフィールドと同じ桁数であって1個のオペコードと1個以上のオペランドとの組が置かれ、

前記VLIWプロセッサはさらに、

前記第3のオペレーションフィールドにオペコードが置かれた場合に前記オペコードを解読する第3の解読手段と、

前記第3の解読手段による解読結果に基づいて前記オペランドによって指定されたデータに対して前記オペコードによって指定されたオペレーションを実行する第3の実行手段とを備えることを特徴とする請求項16記載のVLIWプロセッサ。

【請求項18】 前記第1の実行手段は、実行すべき命令の流れを制御することを特徴とする請求項17記載の

VLIWプロセッサ。

【請求項19】 前記第2の実行手段は、前記第2のオペランドフィールドに置かれたオペランドによって指定されたデータの転送を制御し、

前記第3の実行手段は、前記第3のオペランドフィールドに置かれたオペランドによって指定されたデータの算術論理演算を実行することを特徴とする請求項18記載のVLIWプロセッサ。

【請求項20】 前記フォーマットフィールドの桁数、前記第1のオペレーションフィールドの桁数、前記第2及び第3のオペレーションフィールドに置かれたオペコードの桁数、前記第2及び第3のオペレーションフィールドに置かれた各オペランドの桁数は、いずれもnビットであることとを特徴とする請求項19記載のVLIWプロセッサ。

【請求項21】 前記命令は32ビット長であり、前記nは4であることを特徴とする請求項20記載のVLIWプロセッサ。

【請求項22】 3個以上のオペレーションフィールドを含む命令を解読し実行するデータ処理装置であって、第1の前記オペレーションフィールドには実行すべき命令の流れを制御するオペコードが置かれ、第2の前記オペレーションフィールドにはデータの転送を制御するオペコードが置かれ、

第3の前記オペレーションフィールドにはデータの算術論理演算を制御するオペコードが置かれ、

前記第1のオペレーションフィールドに置かれたオペコードを解読する第1の解読手段と、

前記第1の解読手段による解読結果に基づいて前記オペコードによって指定された実行すべき命令の流れの制御を実行する第1の実行手段と、

前記第2のオペレーションフィールドに置かれたオペコードを解読する第2の解読手段と、

前記第2の解読手段による解読結果に基づいて前記オペコードによって指定されたデータの転送の制御を実行する第2の実行手段と、

前記第3のオペレーションフィールドに置かれたオペコードを解読する第3の解読手段と、

前記第3の解読手段による解読結果に基づいて前記オペコードによって指定されたデータの算術論理演算を実行する第3の実行手段とを備えることを特徴とするVLIWプロセッサ。

【請求項23】 前記第1のオペレーションフィールドの桁数は前記第2及び第3のオペレーションフィールドのいずれの桁数よりも小さいことを特徴とする請求項22記載のVLIWプロセッサ。

【請求項24】 前記第2のオペレーションフィールドの桁数は前記第3のオペレーションフィールドの桁数と等しいことを特徴とする請求項23記載のVLIWプロセッサ。

【請求項25】 前記第1、第2及び第3のオペレーションフィールドに置かれた各オペコードの桁数は等しいことを特徴とする請求項24記載のVLIWプロセッサ。

【請求項26】 3個以上のオペレーションフィールドとフォーマットフィールドを含む命令を解読し実行するデータ処理装置であって、

第1の前記オペレーションフィールドには実行すべき命令の流れを制御するオペコード又は定数が置かれ、

第2の前記オペレーションフィールドにはデータの転送を制御するオペコード又は定数が置かれ、

第3の前記オペレーションフィールドにはデータの算術論理演算を制御するオペコード又は定数が置かれ、

前記フォーマットフィールドには、前記第1、第2及び第3のオペレーションフィールドそれぞれに定数が置かれているか否かを指定するフォーマットコードが置かれ、

前記第1のオペレーションフィールドに置かれたオペコードを解読する第1の解読手段と、

前記第1の解読手段による解読結果に基づいて前記オペコードによって指定された実行すべき命令の流れの制御を実行する第1の実行手段と、

前記第2のオペレーションフィールドに置かれたオペコードを解読する第2の解読手段と、

前記第2の解読手段による解読結果に基づいて前記オペコードによって指定されたデータの転送の制御を実行する第2の実行手段と、

前記第3のオペレーションフィールドに置かれたオペコードを解読する第3の解読手段と、

前記第3の解読手段による解読結果に基づいて前記オペコードによって指定されたデータの算術論理演算を実行する第3の実行手段と、

前記フォーマットコードを解読するフォーマット解読手段と、

前記フォーマット解読手段により前記第1、第2及び第3の少なくとも1つのオペレーションフィールドに定数が置かれていると解読された場合に、その定数を取り出して記憶する定数記憶手段とを備えることを特徴とするVLIWプロセッサ。

【請求項27】 前記命令が32ビット長であることを特徴とする請求項22又は26記載のVLIWプロセッサ。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、VLIWアーキテクチャを採るプロセッサに関し、特に、比較的短い語長であってコード効率の高い命令を実行するプロセッサに関する。

【0002】

【従来の技術】 近年のマルチメディア関連機器の需要の

増大と電子機器の小型化に伴い、音声や画像データ等のマルチメディアデータを高速に処理できるマイクロプロセッサが必要とされている。この要求に応えるマイクロプロセッサとして、VLIW (Very Long Instruction Word) アーキテクチャを採るプロセッサ (以下、「VLIWプロセッサ」という。) がある。

【0003】 VLIWプロセッサは、内部に複数の演算ユニットを備え、1個のVLIWに置かれた複数のオペレーションを同時並列に実行する。このようなVLIWは、コンパイラによってソースプログラムにおけるオペレーションレベルでの並列性が検出されスケジューリングされた後に生成されたものである。ところが、特に機器組み込み用途においては、プログラムのコードサイズが問題となるために、256ビットの如く長いVLIWや、無動作命令 (以下、「NOP命令」という。) が頻繁に挿入されたコード効率の悪いVLIWは好ましくない。

【0004】 比較的短い語長の命令を実行する従来のVLIWプロセッサとして、最大2個のオペレーションを同時に指定することができる32ビットの命令を実行するVLIWプロセッサがある (例えば、特開平9-26878に開示されたデータ処理装置)。図15(a)及び図15(b)は、上記従来技術における命令フォーマットを示し、それぞれ、同時に2個のオペレーションを指定する命令フォーマット、1個のオペレーションだけを指定する命令フォーマットを示す。この従来技術は、2ビットのフォーマットフィールド410の値によってその命令に置かれたオペレーションの数や実行順序を制御することで、コード効率を向上せんとするものである。

【0005】

【発明が解決しようとする課題】 しかしながら、上記従来技術では、32ビット長の1個の命令で同時に指定できるオペレーションの数は最高で2個であり、その並列性は充分とは言えない。また、ある長さの語長を超える定数を用いた演算を行わせる場合には、命令のコード効率が悪化してしまうという問題がある。例えば、32ビットの定数をレジスタにセットするためにその定数を2つに分割し、定数の上位16ビットをセットした後に下位16ビットをセットした場合には、それらオペレーションの指定のためだけに2個の32ビット長命令が消費されてしまう。

【0006】 そこで、本発明はかかる問題点に鑑みてなされたものであり、比較的短い語長の命令であって、かつ、同時に多くのオペレーションを指定することができるコード効率のよい構造を有する命令、例えば、32ビット長の命令であれば3個以上のオペレーションを指定することができるような並列性の高い命令を実行するVLIWプロセッサを提供することを第1の目的とする。

【0007】 また、本発明の第2の目的は、比較的短い

語長の命令であって、かつ、比較的長い語長の定数を扱う場合においてもコード効率が低下しにくい構造を有する命令を実行するVLIWプロセッサを提供することである。

【0008】

【課題を解決するための手段】上記第1の目的を達成するために本発明は、複数のオペレーションフィールドからなる命令を実行するVLIWプロセッサにおいて、前記オペレーションフィールドの大きさが不均一であり、かつ前記命令の命令語長は前記命令の持つオペレーションフィールドの数で割り切れないものであることを特徴とする。

【0009】これによって、命令中の全てのオペレーションフィールドが同じ語長でなければならないという制限から解放され、コード効率のよい命令フォーマットを定義することが可能となるので、比較的短い語長の命令であって、かつ、同時に多くのオペレーションを指定することができるコード効率のよい構造を有する命令を実行するVLIWプロセッサが実現される。

【0010】また、上記第2の目的を達成するために本発明は、2個以上のオペレーションフィールドを含む命令を解説し実行するVLIWプロセッサであって、第1の前記オペレーションフィールドにはオペレーションの種類を指定する1個のオペコードのみ又は定数のみが置かれ、第2の前記オペレーションフィールドには1個のオペコードとオペレーションの対象となるデータを指定する1個以上のオペランドとの組又は定数のみが置かれ、前記第1のオペレーションフィールドにオペコードが置かれた場合に前記オペコードを解説する第1の解説手段と、前記第1の解説手段による解説結果に基づいて前記オペコードによって指定されたオペレーションを実行する第1の実行手段と、前記第2のオペレーションフィールドにオペコードが置かれた場合に前記オペコードを解説する第2の解説手段と、前記第2の解説手段による解説結果に基づいて前記オペランドによって指定されたデータに対して前記オペコードによって指定されたオペレーションを実行する第2の実行手段とを備えることを特徴とする。

【0011】これによって、命令中のあるオペレーションフィールドに無駄なコードを置く必要が生じた場合であっても、他のオペレーションで使用される定数で埋めておくことが可能となるので、比較的短い語長の命令であってもコード効率が低下しにくい構造を有する命令を実行するVLIWプロセッサが実現される。

【0012】

【発明の実施の形態】以下、本発明に係るプロセッサの実施の形態について、図面を用いて詳細に説明する。なお、本明細書では「命令」とは本プロセッサが同時並列に解説し実行するコード全体を意味し、「オペレーション」とは本プロセッサが並列に実行できる数値演算、

論理演算、転送、分岐等の処理単位又はその処理単位を指定するためのコードを意味する。

(命令フォーマット) まず、本プロセッサが解説実行する命令の構造について説明する。

【0013】本プロセッサは、VLIWプロセッサであり、32ビット固定長の命令を解説実行する。図1

(a)は、本プロセッサが実行する命令50のフィールド構成を示す図である。図1(b)～図1(d)は16種類の命令フォーマットを示す図であり、そのうち、図1(b)は3オペレーション、図1(c)は2オペレーション、図1(d)は1オペレーションを同時に指定できる命令フォーマットである。

【0014】この命令50は、32ビット固定長であり、4ビットずつに区切られた8個のフィールド(上位よりP0、0フィールド51、P1、0フィールド52、…、P3、2フィールド58)からなる。なお、P2、0フィールド53～P2、2フィールド55のグループをまとめて第1演算フィールド59と呼び、P3、0フィールド56～P3、2フィールド58のグループをまとめて第2演算フィールド60と呼ぶ。

【0015】図1(b)～図1(d)において、“const”は定数であり、これを用いられるオペレーションの種類によっては即値、絶対番地、ディスプレースメント等の数値定数や文字定数を意味する。“op”はオペレーションの種類を指定するオペコードを、“Rs”はソースオペランドとなるレジスタを、“Rd”はディスティネーションオペランドとなるレジスタを、“cc”は本プロセッサが備える専用の32ビットレジスタ(図3に示される定数レジスタ36)の格納値を分岐先の絶対番地又は相対番地(ディスプレースメント)とする分岐オペレーションを指定するオペコードを意味する。

【0016】また、これらコードの直後に添付された数値は、第1演算フィールド59及び第2演算フィールド60のいずれのオペレーションのために用いられるものであるかを示す。例えば、フォーマットコードが“6”である命令フォーマットの場合作れば、P1、0フィールド52に置かれた4ビットの定数“const1”とP2、1フィールド54に置かれた4ビットの定数“const1”とは結合され、8ビットの定数として第1演算フィールド59のオペコード“op1”に対応するソースオペランドになることを意味する。

【0017】また、数値を伴わない定数“const”は、本プロセッサが備える専用の32ビットレジスタ(図3に示される定数レジスタ36)に格納される定数を示す。例えば、フォーマットコードが“0”である命令フォーマットにおけるP1、0フィールド52に置かれた4ビットの定数“const”は、暗黙的に指定された定数レジスタ36に格納される定数である。

【0018】図2は、図1で用いられている3種類のオペコード“cc”、“op1”及び“op2”それぞれ

によって指定される具体的なオペレーションを説明する図である。4ビットのオペコード“cc”は、16種類の分岐オペレーションの中の一つを指定する。1つの分岐オペレーションは、分岐条件と分岐形式によって特定される。分岐条件には、等しい(“eq”)、等しくない(“neq”)、より大きい(“gt”)等がある。分岐形式には、上記定数レジスタ36の格納値を分岐先の絶対番地として分岐する形式(ニーモニック表示において“i”が添付されていないもの)と相対番地として分岐する形式(ニーモニック表示において“i”が添付されているもの)とがある。例えば、オペコード“eqi”は、直前の比較結果が等しい場合に相対番地指定による分岐を行なうオペレーションを意味し、オペコード“eqi”は、直前の比較結果が等しい場合に相対番地指定による分岐を行なうオペレーションの一つを指定する場合と、“mov” (ワード(32ビット)データの転送)、“movh” (ハーフワードデータの転送)、“movb” (バイトデータの転送)等のレジスタ・レジスタ間転送に属するオペレーションの一つを指定する場合とがある。

【0019】4ビットのオペコード“opl”は、“add” (加算)、“sub” (減算)、“mul” (乗算)、“and” (論理積)、“or” (論理和)等の算術論理演算に属するオペレーションの一つを指定する場合と、“mov” (ワード(32ビット)データの転送)、“movh” (ハーフワードデータの転送)、“movb” (バイトデータの転送)等のレジスタ・レジスタ間転送に属するオペレーションの一つを指定する場合とがある。

【0020】4ビットのオペコード“op2”は、上記オペコード“opl”と同様の算術論理演算及びレジスタ・レジスタ間転送に加えて、“ld” (メモリからレジスタへの1ワードデータのロード)、“st” (レジスタからメモリへのワードデータのストア)等のレジスタ・メモリ間転送に属するオペレーションの一つを指定する場合がある。

【0021】次に、図1(a)に示された各フィールド51、52、59、60の特徴を説明する。P0、0フィールド51は、この命令50のフォーマットを特定する4ビットのフォーマットコードを置くためのフィールドであり、具体的には、図1(b)～図1(d)に示された16種類の命令フォーマットの一つを特定する。

【0022】P1、0フィールド52は、定数又は分岐用のオペコードを置くためのフィールドである。このP1、0フィールド52に定数が置かれた場合(フォーマットコード=0、1、4～9の場合)には、その定数は、定数レジスタ36に格納する対象となる場合(フォーマットコード=0、1、4、5の場合)と、第1演算フィールド59又は第2演算フィールド60のオペランドの一部を構成する場合(フォーマットコード=5、7、8、9、Bの場合)とがある。さらに、定数レジスタ36に格納する対象となる場合には、その4ビットの定数のみが格納される場合(フォーマットコード=0、1の場合)と、第1演算フィールド59又は第2演算フィールド60に置かれた12ビットの定数と共に格納さ

れる場合(フォーマットコード=4、5の場合)とがある。

【0023】一方、このP1、0フィールド52に分岐用のオペコード“cc”が置かれた場合(フォーマットコード=2、3、Aの場合)には、定数レジスタ36に格納された定数を分岐先の絶対番地として、又は、相対番地(ディスプレースメント)として分岐すること意味する。第1演算フィールド59は、本プロセッサと外部(メモリ)とのデータの転送を伴わないオペレーション(算術論理演算、レジスタ間転送)を指定するためのオペコードとオペランド(ソース及びデスティネーション)との組又は定数が置かれる。

【0024】第2演算フィールド60は、上記第1演算フィールド59の場合に加えて、本プロセッサと外部(メモリ)とのデータの転送を伴うオペレーション(レジスタ・メモリ間転送)を指定するためのオペコードとオペランドとの組が置かれることもある。なお、以上のようなオペレーションの種類の各フィールドへの割当ては、ノーマン型の本プロセッサにおいては2つ以上の分岐オペレーションを同時に実行する必要がないこと、本プロセッサと外部(メモリ)とのオペランドの入出力ポート(図3におけるオペランドアクセス部40)を1つに限定していること等に基づく。

【0025】ここで、図1(b)～図1(d)に示された命令フォーマットには以下の特徴がある。第1に、定数“const”に着目して判るように、定数レジスタ36に定数を格納させる命令フォーマットは次の3通りである。

(1) フォーマットコードが“0”又は“1”の場合: この命令では、P1、0フィールド52に置かれた4ビットの定数が定数レジスタ36に格納される。

(2) フォーマットコードが“4”の場合: この命令では、P1、0フィールド52～P2、2フィールド55に置かれた16ビットの定数が定数レジスタ36に格納される。

(3) フォーマットコードが“5”の場合: この命令では、P1、0フィールド52とP3、0フィールド56～P3、2フィールド58に置かれた16ビットの定数が定数レジスタ36に格納される。

【0026】第2に、本プロセッサでは、1個の命令に最大3つのオペレーションを指定することができるが、その場合には、図1(b)に示された3オペレーション用の命令フォーマットから判るように、それら3つのオペレーションの種類の組み合わせになる。

(1) 4ビットの定数を定数レジスタ36にセットするオペレーションと2個の汎用オペレーション(フォーマットコードが“0”、“1”の場合)

(2) 定数レジスタ36にセットされた値を絶対番地又は相対番地として分岐するオペレーションと2個の汎用

オペレーション（フォーマットコードが“2”、“3”の場合）

このように、本プロセッサの命令は、わずか32ビット長でありながら最大3つのオペレーションを同時に指定することができるコード効率の高いフィールド構成を有している。

（プロセッサのハードウェア構成）次に、本プロセッサのハードウェア構成を説明する。

【0027】図3は、本発明に係るプロセッサのハードウェア構成を示すブロック図である。本プロセッサは、上述したように、最大3つのオペレーションを並列実行するVLIWプロセッサであり、大きく分けて、命令レジスタ10、解読部20及び実行部30から構成される。

【0028】命令レジスタ10は、命令フェッチ部39から送られてきた1個の命令を保持する32ビットのレジスタである。解読部20は、命令レジスタ10に保持された命令を解読し、その解読結果に応じた制御線を実行部30に出力するものであり、大きく分けて、フォーマットデコード21と命令デコード22とからなる。

【0029】命令デコード22はさらに、P1、0フィールド12に保持されたオペコード“c”を解読しその結果に基づいてPC部33を制御する分岐デコード23と、P2、0フィールド13に保持されたオペコードを解読しその結果に基づいて第1演算部37を制御する第1演算デコード24と、P3、0フィールド16に保持されたオペコードを解読しその結果に基づいて第2演算部38及びオペランドアクセス部40を制御する第2演算デコード25とからなる。

【0030】フォーマットデコード21は、P0、0フィールド11に保持された4ビットのフォーマットコードをデコードすることによって命令レジスタ10に保持された命令のフォーマットが図1（b）～図1（d）に示された16種類のうちのいずれであるかを特定し、その結果に応じて分岐デコード23、第1演算デコード24及び第2演算デコード25による解読動作を許可又は禁止とし、実行部30の定数レジスタ制御部32を動作させたりする。

【0031】なお、上記デコード21、23～25は、基本的には1サイクルに1つのオペレーションを解読し、実行部30に制御信号を与える。また、命令レジスタ10と実行部30を接続する26ビットの定数信号線26は、命令レジスタ10に置かれた定数やオペランドを実行部30に転送するためのバスである。実行部30は、解読部20での解読結果に基づいて、最大3つのオペレーションを並列実行する回路ユニットであり、実行制御部31、PC部33、レジスタ群34、第1演算部37、第2演算部38、命令フェッチ部39及びオペランドアクセス部40からなる。なお、この実行部30のうち定数レジスタ制御部32、PC部33及び定数レジ

スタ36については、別の図面においてさらに詳細な構成を示している。

【0032】実行制御部31は、解読部20での実行結果に基づいて実行部30の各構成要素3～40を制御する制御回路や配線の総称であり、通常のプロセッサが備える構成要素（タイミング制御、動作許可禁止制御、ステータス管理、割り込み制御等の回路）の他に本プロセッサに特有の定数レジスタ制御部32を有する。定数レジスタ制御部32は、フォーマットデコード21からの指示に基づいて命令レジスタ10に保持された4ビット又は16ビットの定数（const）を定数レジスタ36に格納する制御を行なう。

【0033】PC（プログラムカウンタ）部33は、分岐デコード23による制御の下で、次に解読実行すべき命令が置かれている図示されていない外部メモリ上のアドレスを命令フェッチ部39に出力する。命令フェッチ部39は、32ビットのIA（インストラクションアドレス）バス及び32ビットのID（インストラクションデータ）バスを通じて図示されていない外部メモリから命令ブロックをフェッチし、内部の命令キャッシュに保持すると共に、PC部33から出力されたアドレスに相当する命令を命令レジスタ10に供給する。

【0034】レジスタ群34は、15個の32ビット汎用レジスタ35と1個の32ビット定数レジスタ36から構成される。これら16個のレジスタ35、36に格納された値は、第1演算デコード24及び第2演算デコード25での解読結果に基づいて、第1演算部37及び第2演算部38に転送され、ここで演算が施され、又は、ここを単に通過した後に、レジスタ群34又はオペランドアクセス部40に送られる。なお、定数レジスタ36に格納された値は、第1演算部37及び第2演算部38での演算に用いられる他に、PC部33にも転送され、ここで分岐先となる有効アドレスを生成するために用いられる。

【0035】第1演算部37は、2個の32ビットデータに対して算術論理演算を行なうALUと乗算を行う乗算器とを内部に有し、第1演算デコード24による制御の下で2種類のオペレーション（算術論理演算とレジスタ間転送）を実行する。第2演算部38も、第1演算部37と同様に、2個の32ビットデータに対して算術論理演算を行なうALUと乗算を行う乗算器とを内部に有し、第2演算デコード25による制御の下で2種類のオペレーション（算術論理演算とレジスタ間転送）を実行する。

【0036】オペランドアクセス部40は、第2演算デコードによる制御の下でレジスタ群34と図示されていない外部メモリとの間でオペランドの転送を行なう回路であり、そのオペランドやオペランドアドレスを保持するバッファを内部に有する。具体的には、例えば、命令レジスタ10のP3、1フィールド16にオペコード



“1d”が置かれていた場合には、外部メモリに置かれていた1ワードのデータがオペランドアクセス部40を経てレジスタ群34のいずれかのレジスタにロードされ、また、オペコード“st”が置かれていた場合には、レジスタ群34のいずれかのレジスタの格納値が外部メモリにストアされる。

【0037】上記PC部33、レジスタ群34、第1演算部37、第2演算部38及びオペランドアクセス部40は、図示されるように、内部バス(L1バス、R1バス、L2バス、R2バス、D1バス、D2バス)で接続されている。なお、L1バス及びR1バスはそれぞれ第1演算部37の2つの入力ポートに、L2バス及びR2バスはそれぞれ第2演算部38の2つの入力ポートに、D1バス及びD2バスはそれぞれ第1演算部37及び第2演算部38の出力ポートに接続されている。

(定数レジスタ36及びその周辺回路の詳細な構成)次に、定数レジスタ36及びその周辺回路について詳細に説明する。

【0038】図4は、定数レジスタ36及びその周辺回路の詳細な構成と接続関係を示すブロック図である。なお、図中の固定値(“0”)27は、定数“0”を示す4本の信号線の固定的な配線を意味する。定数レジスタ制御部32は、5個の3入力セレクト32a~32eと3個の4入力セレクト32f~32hとからなり、定数レジスタ36は、8個の4ビット幅レジスタ36a~36hからなる。なお、各入出力データは並列4ビットである。

【0039】定数レジスタ制御部32は、フォーマットデコーダ21及び命令デコーダ22からの制御信号に従って上記8個の入力セレクト32a~32hを制御することで、以下に示す4通りの格納方法のいずれかの方法により、命令レジスタ10に保持された定数又はゼロを定数レジスタ36に格納させる。図5(a)~図5(d)は、その4通りの格納方法を説明する図である。

【0040】図5(a)は、フォーマットデコーダ21によってP0、0フィールド11に保持された値が“0”又は“1”であるとして読取されたデータの格納方法を示す。これは、P1、0フィールド12に置かれた4ビットの定数のみを定数レジスタ36に格納する場合に相当する。具体的には、定数レジスタ36に保持されたデータを4ビット単位で上位にシフトさせると同時に、命令レジスタ10のP1、0フィールド12に保持された4ビットの定数を定数レジスタ36の最下位の4ビットレジスタ36hに格納する。

【0041】図5(b)は、フォーマットデコーダ21によってP0、0フィールド11に保持された値が“4”であるとして読取されたデータの格納方法を示す。これは、P1、0フィールド12~P2、2フィールド15に置かれた16ビットの定数を定数レジスタ36に格納する場合に相当する。具体的には、定数レジスタ36の

下位16ビット36e~36hに保持されたデータを上位16ビット36a~36dにシフトさせると同時に、命令レジスタ10のP1、0フィールド12~P2、2フィールド15に保持された16ビットの定数を定数レジスタ36の下位16ビット36e~36hに格納する。

【0042】図5(c)は、フォーマットデコーダ21によってP0、0フィールド11に保持された値が

“5”であるとして読取されたデータの格納方法を示す。これは、P1、0フィールド12とP3、0フィールド16~P3、2フィールド18に置かれた16ビットの定数を定数レジスタ36に格納する場合に相当する。具体的には、定数レジスタ36の下位16ビット36e~36hに保持されたデータを上位16ビット36a~36dにシフトさせると同時に、命令レジスタ10のP1、0フィールド12とP3、0フィールド16~P3、2フィールド18に保持された16ビットの定数を定数レジスタ36の下位16ビット36e~36hに格納する。

【0043】図5(d)は、フォーマットデコーダ21によってP0、0フィールド11に保持された値が

“2”、“3”及び“A”のいずれかであると読取された場合又は命令デコーダ22によってP2、1フィールド14、P2、2フィールド15、P3、2フィールド17及びP3、3フィールド18の少なくとも一つに定数レジスタ(R15)が指定されている場合の格納方法を示す。これは、P1、0フィールド12に置かれた分岐オペレーション、第1演算フィールド59及び第2演算フィールド60の少なくとも一つとのオペレーションによって定数レジスタ36の格納値が使用された(読み出された)後に、定数レジスタ36にオールゼロを格納する(定数レジスタ36をクリアする)場合に相当する。

【0044】具体的には、定数レジスタ36の格納値がPC部33、第1演算部37及び第2演算部38のいずれかに読み出された直後に、32ビットの定数“0”を定数レジスタ36に格納する。なお、定数レジスタ36の使用後にクリアしておくのは、定数レジスタ36には常にゼロ拡張された値が格納されていることを保証するためである。ここで、ゼロ拡張とは、ある数値の有効桁数が一定の桁数に満たない場合に、その有効桁より上位の桁全てをゼロで埋める処理をいう。

【0045】以上のように、命令レジスタ10のP0、0フィールド11の値が“0”、“1”、“4”、“5”の場合には、定数レジスタ36に既に格納された定数をシフトさせながら新たな定数が定数レジスタ36に格納される。また、定数レジスタ36は、その格納値が一旦読み出されて使用されると、その内容は消去される。このようにして、定数レジスタ36、その内容が読み出されるまで、次々に格納される定数を蓄積していくことができる。

(PC部33の詳細な構成)次に、PC部33の詳細な

構成を説明する。

【0046】図6は、PC部33の詳細な構成を示すブロック図である。PC部33は、定数“4”を示す固定的な配線である固定値（“4”）33a、2入力セレクト33b、加算器33c、次に読読実行すべき命令のアドレスを保持するPC33d及び4入力セレクト33eから構成される。このPC部33では、読読部20からの制御信号に従ってセレクト33b、33eが動作することにより、以下の3通りの値のいずれかが有効アドレスとしてセレクト33eから命令フェッチ部39に出力される。

(1) PC33dの内容に“4”を加算した値

これは、分岐しないで順次に行なう場合、即ち、読読実行された命令に分岐オペレーションが指定されていない場合に相当する。なお、“4”を加算するのは、1つの命令の長さが4バイト（32ビット）であることによる。

(2) PC33dの内容に定数レジスタ36の内容を加算した値

これは、定数レジスタ36の内容を相対番地として分岐する場合、例えば、P1.0フィールド12によって相対番地による分岐が指定されていると分岐デコード23が読読した場合は該当する。

(3) 定数レジスタ36の内容

これは、定数レジスタ36の内容を絶対番地として分岐する場合、例えば、P1.0フィールド12によって絶対番地による分岐が指定されていると分岐デコード23が読読した場合は該当する。

【0047】以上のように、このPC部33は、専用の加算器33cを備え、定数レジスタ36に保持された値を直接用いる構成となっているので、第1演算部37や第2演算部38での演算とは独立並行して、定数レジスタ36の格納値を絶対番地又は相対番地として分岐する実行制御を行なうことができる。

（プロセッサの動作）次に、具体的な命令を読読実行した場合の本プロセッサの動作について説明する。

【0048】図7は、32ビットの定数を扱う処理の一例を示すフローチャートである。本図には、レジスタR0とR1との格納値の差を求め（ステップS80）、その結果にレジスタR2の格納値を掛け（ステップS81）、さらにその結果に32ビットの定数“0x87654321”（16進数の“87654321”）を加え（ステップS82、S83）、最後にレジスタR3をクリアしておく（ステップS85）という処理が示されている。

【0049】図8は、図7に示された処理内容を本プロセッサに行なわせるプログラムの例を示す図である。このプログラムは、3個の命令71～73から構成されている。1行が1個の命令に相当し、各命令の内容は各フィールドに置かれたニーモニックで表現されている。な

お、定数は全て16進数で表現されている。また、“f m t n (n=0～F)”はフォーマットコード“n”を示し、“R n (n=0～15)”はレジスタ群34の中の1つのレジスタを示す。なお、“R15”は定数レジスタ36を意味する。

【0050】図9は、図8に示されたプログラムを実行した場合の本プロセッサの動作を示すタイミングチャートである。本図には、クロックサイクル、汎用レジスタR0～R3及び定数レジスタR15の内容、4つのバスL1、R1、L2、R2を流れるデータが示されている。上記図8及び図9を用いて、各命令71～73ごとの本プロセッサの動作を説明する。

（命令71）命令71が命令レジスタ10にロードされると、本プロセッサは図9のクロックサイクル0～t1に示された動作をする。

【0051】フォーマットデコード21は、命令レジスタ10のP0.0フィールド11の値（“f m t 4”）から、この命令はフォーマットコードが“4”の2オペレーション命令であると判断し、以下の2つのオペレーションが並列実行されるように実行部30を制御する。

(1) 第1のオペレーション

定数レジスタ制御部32は、内部の8個のセレクト32a～32hを制御することで、図5(b)に示された格納方法により、P1.0フィールド12～P2.2フィールド15に保持された16ビットの定数（0x8765）を定数レジスタ36の下位16ビットに格納する。その結果、図9のクロックサイクル0～t1に示されるように、定数レジスタR15の内容は、それまでの“0x00000000”から“0x00008765”に変化する。

(2) 第2のオペレーション

第2演算部38は、汎用レジスタR0の内容（“0x33333333”）と汎用レジスタR1の内容（“0x22222222”）とを入力とし、ここで減算した後に、その結果を再び汎用レジスタR0に格納する。その結果、図9のクロックサイクル0～t1に示されるように、汎用レジスタR0の内容は、それまでの“0x33333333”から“0x11111111”に変化する。

（命令72）次に、命令72が命令レジスタ10にロードされると、本プロセッサは図9のクロックサイクルt1～t2に示された動作をする。

【0052】フォーマットデコード21は、上記命令71の場合と同様に、命令レジスタ10のP0.0フィールド11の値（“f m t 4”）から、この命令はフォーマットコードが“4”の2オペレーション命令であると判断し、以下の2つのオペレーションが並列実行されるように実行部30を制御する。

(1) 第1のオペレーション

定数レジスタ制御部32は、内部の8個のセレクト32

a~32hを制御することで、図5(b)に示された格納方法により、P1.0フィールド12~P2.2フィールド15に保持された16ビットの定数(0x4321)を定数レジスタ36の下位16ビットに格納する。その結果、図9のクロックサイクルt1~t2に示されるように、定数レジスタR15の内容は、それまでの“0x00008765”から“0x87654321”に変化する。

(2) 第2のオペレーション  
第2演算部38は、汎用レジスタR2の内容(“0x00000004”)と汎用レジスタR0の内容(“0x11111111”)とを入力とし、ここで乗算した後に、その結果を再び汎用レジスタR0に格納する。その結果、図9のクロックサイクルt1~t2に示されるように、汎用レジスタR0の内容は、それまでの“0x11111111”から“0x44444444”に変化する。

(命令73) 最後に、命令73が命令レジスタ10にロードされると、本プロセッサは図9のクロックサイクルt2~t3に示された動作をする。

【0053】フォーマットデコード21は、命令レジスタ10のP0.0フィールド11の値(“fmt7”)から、この命令はフォーマットコードが“7”の2オペレーション命令であると判断し、以下の2つのオペレーションが並列実行されるように実行部30を制御する。

(1) 第1のオペレーション

第1演算部37は、定数レジスタR15の内容(“0x87654321”)と汎用レジスタR0の内容(“0x44444444”)とを入力とし、それらを加算した後に、その結果を再び汎用レジスタR0に格納する。その結果、図9のクロックサイクルt2~t3に示されるように、汎用レジスタR0の内容は、それまでの“0x44444444”から“0xCBA98765”に変化し、定数レジスタR15の内容はクリアされる。

(2) 第2のオペレーション

第2演算部38は、P1.0フィールド12とP3.1フィールド17に分割して置かれた8ビットの定数(“0x00”)を入力とし、そのまま通過させて、汎用レジスタR3に格納する。その結果、図9のクロックサイクルt2~t3に示されるように、汎用レジスタR3の内容は、それまでの“0xFEDCBA98”から“0x00000000”に変化する。

【0054】以上のようにして、本プロセッサにおいて、32ビットの定数“0x87654321”は、2個の命令71、72に跨って分割配置され、順次定数レジスタ36にシフトされながら格納された後に、第3番目の命令73によって利用された。このようにして、図7のフローチャートに示された処理が3個の命令71~73によって実行される。次に、16ビットの定数を扱

う別のプログラムを用いて本プロセッサの動作を説明する。

【0055】図10は、16ビットの定数を扱うプログラムの例を示す図である。このプログラムは、5個の命令74~78から構成されている。各命令71~73ごとの本プロセッサの動作は以下の通りである。

(命令74) 命令74が命令レジスタ10にロードされると、フォーマットデコード21は、命令レジスタ10のP0.0フィールド11の値(“fmt0”)から、この命令はフォーマットコードが“0”の3オペレーション命令であると判断し、以下の3つのオペレーションが並列実行されるように実行部30を制御する。

(1) 第1のオペレーション

定数レジスタ制御部32は、内部の8個のセクタ32a~32hを制御することで、図5(a)に示された格納方法により、P1.0フィールド12に保持された4ビットの定数(“0x8”)を定数レジスタ36の最下位の4ビットレジスタ36hに格納する。

(2) 第2のオペレーション

第1演算部37は、汎用レジスタR6の値を入力とし、そのまま通過させて、汎用レジスタR1に格納する。

(3) 第3のオペレーション

同様に、第2演算部38は、汎用レジスタR7の値を入力とし、そのまま通過させて、汎用レジスタR2に格納する。

(命令75) 同様にして、命令75が命令レジスタ10にロードされると、フォーマットデコード21は、この命令はフォーマットコードが“0”の3オペレーション命令であると判断し、以下の3つのオペレーションが並列実行されるように実行部30を制御する。

(1) 第1のオペレーション

定数レジスタ制御部32は、内部の8個のセクタ32a~32hを制御することで、図5(a)に示された格納方法により、P1.0フィールド12に保持された4ビットの定数(“0x7”)を定数レジスタ36の最下位4ビットレジスタ36hに格納する。この結果、定数レジスタ36の下位8ビットには定数“0x87”がセットされる。

(2) 第2のオペレーション

第1演算部37は、汎用レジスタR0とR1の値を入力とし、ここで加算した後に、その結果を再び汎用レジスタR1に格納する。

(3) 第3のオペレーション

同様に、第2演算部38は、汎用レジスタR0とR2の値を入力とし、ここで加算した後に、その結果を再び汎用レジスタR2に格納する。

(命令76、命令77) 同様にして、命令76、77が実行されることにより、定数レジスタ36の下位16ビットには定数“0x8765”がセットされる。

(命令78) 命令78が命令レジスタ10にロードされ

ると、本プロセッサは、図8に示された命令73の場合と同様の動作をする。

【0056】以上のようにして、本プロセッサにおいては、16ビットの定数“0x8765”は、4個の命令74~77に跨って分割配置され、順次定数レジスタ36にシフトされながら格納された後に、第5番目の命令78によって利用された。

(通常のプロセッサとの比較) 次に、上記図8及び図10に示されたプログラムと同一内容の処理を通常のプロセッサに行なわせる場合について説明し、本発明に係るプロセッサと比較する。なお、ここでいう通常のプロセッサとは、本発明に係るプロセッサの定数レジスタ36や定数レジスタ制御部32の如く、分割された定数を蓄積して格納する手段のみを有しないプロセッサをいい、32ビット固定長の命令を実行するものとする。

【0057】図11(a)は、この通常のプロセッサが実行する命令のフィールド定義を示し、図11(b)は、その命令のフォーマットを示す。つまり、通常のプロセッサは、3種類の2オペレーション命令101~103と1種類の1オペレーション命令104を実行するものとする。図12は、図8に示されたプログラムと同一内容の処理、即ち、図7のフローチャートに示された処理を通常のプロセッサに行なわせるプログラムの例である。

【0058】図12と図8とを比較して判るように、通常のプロセッサ用のプログラムは、本発明に係るプロセッサ用のものよりも2個の命令だけ多くなっている。なお、命令105、106にnopコードが含まれるのは、命令106は命令105での演算結果を用いるので、これらの命令を並列に実行させることができないからである。また、1個の定数“0x87654321”を上位16ビットと下位16ビットの2つに分割して定数レジスタR1にセットしているのは(命令107、108)、32ビットの1個の命令の中に、セット命令のオペコードと32ビットの定数の両方を同時に配置することは不可能だからである。

【0059】同様に、図13は、図10に示されたプログラムと同一内容の処理を通常のプロセッサに行なわせるプログラムの例である。図13と図10とを比較して判るように、通常のプロセッサ用のプログラムは、本発明に係るプロセッサ用のものよりも1個の命令だけ多くなっている。以上のように、本発明に係るプロセッサが実行する命令は、32ビットという比較的短い語長でありながら最大3つのオペレーションを同時に指定することができるコード効率の高いフィールド構成を有している。

【0060】そして、本発明に係るプロセッサによれば、16ビットや32ビットの定数が複数の命令に跨って分割配置されていても、それらは定数レジスタ36に蓄積して格納されることで元の定数に復元され、分岐や

算術演算等のオペレーションに使用される。つまり、命令中に生じた小さな領域であっても、定数を分割して埋めておくことができるので、通常のプロセッサに実行させる場合よりもプログラムのコードサイズは縮小される。以上、本発明に係るプロセッサについて、実施形態に基づいて説明したが、本発明はこれら実施形態に限らないことは勿論である。即ち、

(1) 上記実施の形態では、命令50は、32ビット長であり、8個の4ビット長のフィールドからなり、最大3個のオペレーションを指定することができる構造を有したが、本発明は、これら数値に限定されるものではない。

【0061】例えば、上記フィールド構成にさらに、1個の4ビット長のオペコードと1個の4ビット長のオペランドとの組からなる8ビット長のフィールドを加えた合計40ビット長の命令とすることもできる。これによって、40ビットという比較的短い語長の命令でありながら最大4つのオペレーションを同時に実行させることができるコード効率の高い命令が定義される。

(2) また、32ビット固定長命令によって3個のオペレーションを同時に指定することができる命令の構造として、図1(a)に示された命令構造の他に、図14(a)~図14(d)に示された命令構造とすることもできる。これら図中において、縦線の最小間隔は1ビット長を示し、“fmt”はフォーマットフィールドを示す。

【0062】図14(a)に示された構造の命令であれば、上記実施形態の命令に比較し、より多くの命令フォーマットを定義することができる点、及び、3つのオペレーションフィールドそれぞれに少なくとも1個のオペランドを置くことができる点において優る。図14(b)~図14(d)に示された構造の命令であれば、上記実施形態の命令に比較し、2個のオペコード(“op2”、“op3”)の桁数が大きいので、より多くの種類のオペレーションを定義することができる点において優る。

(3) また、上記実施の形態の命令50では、暗黙的なオペランド(定数レジスタ36の格納値)を用いるフィールドは1箇所だけであったが、これに限定されるものではなく、2箇所以上であってもよい。新たな命令フォーマットを定義することで対応すればよい。

(4) また、上記実施の形態では、数値定数を扱う例が示されたが、文字定数であってもよいことは言うまでもない。複数の命令に跨って分割配置された文字定数であっても、定数レジスタ36への複数回の格納によって、桁数の長い元の文字定数が復元されるからである。

(5) また、上記実施の形態では、図1(b)~図1(d)の命令フォーマットから判るように、1個の命令によって定数レジスタ36に格納させることができる定数の桁数は4ビット及び16ビットのいずれかであった

が、本発明はこの桁数に限定されるものではない。例えば、12ビットや28ビットの定数を定数レジスタ36に格納するための命令フォーマットを定義してもよい。そのためには、定数レジスタ36の周辺回路の接続関係を変更すればよい。

#### 【0063】

【発明の効果】以上の説明から明らかなように、本発明は、複数のオペレーションフィールドからなる命令を実行するVLIWプロセッサにおいて、前記オペレーションフィールドの大きさが不均一であり、かつ前記命令の命令語長は前記命令の持つオペレーションフィールドの数で割り切れないものであることを特徴とする。

【0064】これによって、命令中の全てのオペレーションフィールドが同じ語長でなければならないという制限から解放され、コード効率のよい命令フォーマットを定義することが可能となるので、比較的短い語長の命令であっても、かつ、同時に多くのオペレーションを指定することができるコード効率のよい構造を有する命令を実行するVLIWプロセッサが実現される。

【0065】ここで、複数のオペレーションフィールドからなる命令を実行するVLIWプロセッサにおいて、前記オペレーションフィールドの大きさが不均一であり、かつ前記命令は3つのオペレーションフィールドを32ビットの命令語長中に持つものであるとすることもできる。これによって、32ビット長で3個のオペレーションを指定することができる並列性の高い命令を実行するVLIWプロセッサが実現される。

【0066】また、複数のオペレーションフィールドからなる命令を実行するVLIWプロセッサにおいて、前記オペレーションフィールドのうち少なくとも1つはオペランドの数が異なるものであるとすることもできる。これによって、命令中の全てのオペレーションフィールドが同じ個数のオペランドを有さなければならないという制限から解放されるので、コード効率のよい命令フォーマットを定義することが可能となる。

【0067】また、複数のオペレーションフィールドからなる命令を実行するVLIWプロセッサにおいて、前記オペレーションフィールドのうち1つはオペコードのみからなるとすることもできる。これによって、命令中の全てのオペレーションフィールドにオペコードとオペランドとの組が置かれる場合に比べ、命令の語長は短縮されるので、コード効率のよい構造を有する命令を実行するVLIWプロセッサが実現される。

【0068】また、2個以上のオペレーションフィールドを含む命令を解釈し実行するVLIWプロセッサであって、第1の前記オペレーションフィールドにはオペレーションの種類を指定する1個のオペコードのみが置かれ、第2の前記オペレーションフィールドには1個のオペコードとオペレーションの対象となるデータを指定する1個以上のオペランドとの組が置かれ、前記第1のオ

ペレーションフィールドに置かれたオペコードを解釈する第1の解釈手段と、前記第1の解釈手段による解釈結果に基づいて前記オペコードによって指定されたオペレーションを実行する第1の実行手段と、前記第2のオペレーションフィールドに置かれたオペコードを解釈する第2の解釈手段と、前記第2の解釈手段による解釈結果に基づいて前記オペランドによって指定されたデータに対して前記オペコードによって指定されたオペレーションを実行する第2の実行手段とを備えるとしてもできる。

【0069】これによって、命令中の少なくとも1つのオペレーションには明示的なオペランドを伴わないオペコードのみを置くことができるので、命令語長は短縮される。また、前記第1のオペレーションフィールドに置かれたオペコードの桁数は前記第2のオペレーションフィールドに置かれたオペコードの桁数と等しいとすることもできる。

【0070】これによって、命令中に置かれる全てのオペコードの桁数を共通にすることができるので、デコーダ回路等が簡単化される。また、前記命令に含まれるオペレーションフィールドは3個であり、第3の前記オペレーションフィールドは前記第2のオペレーションフィールドと同じ桁数であって1個のオペコードと1個以上のオペランドとの組が置かれ、前記VLIWプロセッサはさらに、前記第3のオペレーションフィールドにオペコードが置かれた場合に前記オペコードを解釈する第3の解釈手段と、前記第3の解釈手段による解釈結果に基づいて前記オペランドによって指定されたデータに対して前記オペコードによって指定されたオペレーションを実行する第3の実行手段とを備えるとしてもできる。

【0071】これによって、同時に3個のオペレーションを実行するVLIWプロセッサが実現される。また、前記第1の実行手段は、実行すべき命令の流れを制御するとしてもできる。これによって、一般的に多くの桁数を必要としない分岐オペレーションが桁数の小さいオペレーションフィールドに割り当てられるので、コード効率のよい命令セットが定義される。

【0072】また、前記第2の実行手段は、前記第2のオペランドフィールドに置かれたオペランドによって指定されたデータの転送を制御し、前記第3の実行手段は、前記第3のオペランドフィールドに置かれたオペランドによって指定されたデータの算術論理演算を実行するとしてもできる。これによって、外部メモリとのデータ転送は命令中の1個のオペレーションだけによって指定されることとなるので、VLIWプロセッサが備えるべきオペランドアクセス回路は簡単化される。

【0073】また、2個以上のオペレーションフィールドを含む命令を解釈し実行するVLIWプロセッサであって、第1の前記オペレーションフィールドにはオペ

ーションの種類を指定する1個のオペコードのみ又は定数のみが置かれ、第2の前記オペレーションフィールドには1個のオペコードとオペレーションの対象となるデータを指定する1個以上のオペランドとの組又は定数のみが置かれ、前記第1のオペレーションフィールドにオペコードが置かれた場合に前記オペコードを解読する第1の解読手段と、前記第1の解読手段による解読結果に基づいて前記オペコードによって指定されたオペレーションを実行する第1の実行手段と、前記第2のオペレーションフィールドにオペコードが置かれた場合に前記オペコードを解読する第2の解読手段と、前記第2の解読手段による解読結果に基づいて前記オペランドによって指定されたデータに対して前記オペコードによって指定されたオペレーションを実行する第2の実行手段とを備えることもできる。

【0074】これによって、命令中のあるオペレーションフィールドに無駄なコードを置く必要が生じた場合であっても、他のオペレーションで使用される定数で埋めておくことが可能となるので、比較的短い語長の命令であってもコード効率が悪化しにくい構造を有する命令を実行するVLIWプロセッサが実現される。また、前記命令はさらに、前記第1及び第2のオペレーションフィールドそれぞれに定数のみが置かれているか否かを指定するフォーマットコードが置かれたフォーマットフィールドを含み、前記VLIWプロセッサはさらに、前記フォーマットコードを解読するフォーマット解読手段と、前記フォーマット解読手段により前記第1、第2及び第3の少なくとも1つのオペレーションフィールドに定数のみが置かれていると解読された場合に、その定数を取り出して記憶する定数記憶手段とを備えることもできる。

【0075】これによって、オペレーションフィールドに埋められた定数は定数記憶手段に格納され、他の命令中に置かれたオペレーションによってその定数を利用することが可能となるので、比較的短い語長の命令であっても、かつ、比較的長い語長の定数を扱う場合においてもコード効率の低下が回避される。また、前記フォーマットフィールドの桁数、前記第1のオペレーションフィールドの桁数、前記第2及び第3のオペレーションフィールドに置かれたオペランドの桁数は、いずれもnビットであるとしてもできる。

【0076】これによって、1個の命令を構成する全てのフィールドの桁数が同じになるので、VLIWプロセッサの内部回路が簡素化される。以上のように、本発明によって、比較的短い語長の命令であって、かつ、同時に多くのオペレーションを指定することができるコード効率のよい構造を有する命令を実行するVLIWプロセッサが実現され、特にマルチメディアデータを処理する組み込み用途のプロセッサとしてその実用的価値は大き

い。

#### 【図面の簡単な説明】

【図1】図1(a)は、本発明に係るプロセッサが実行する命令のフィールド構成を示す図である。図1(b)～図1(d)は、16種類の命令フォーマットを示す図である。図1(b)は3オペレーション、図1(c)は2オペレーション、図1(d)は1オペレーションを同時に指定できる命令フォーマットである。

【図2】図1で用いられている3種類のオペコード“cc”、“op1”及び“op2”それぞれによって指定される具体的なオペレーションを示する図である。

【図3】同プロセッサのハードウェア構成を示すブロック図である。

【図4】同プロセッサの定数レジスタ36及びその周辺回路の詳細な構成を示すブロック図である。

【図5】図4に示された定数レジスタ制御部32による定数の格納方法を示す図である。図5(a)はフォーマットコードが“0”又は“1”である場合、図5(b)はフォーマットデコードが“4”である場合、図5(c)はフォーマットデコードが“5”である場合、図5(d)はフォーマットコードが“2”、“3”及び“A”のいずれかである場合又は定数レジスタ36の格納値がオペランドとして指定されている場合の格納方法を示す。

【図6】同プロセッサのPC部33の詳細な構成を示すブロック図である。

【図7】32ビットの定数を扱う処理の一例を示すフローチャートである。

【図8】図7に示された処理を同プロセッサに行なわせるプログラムの例を示す図である。

【図9】図9は、図8に示されたプログラムを実行した場合の本プロセッサの動作を示すタイミングチャートである。

【図10】16ビットの定数を扱う処理を同プロセッサに行なわせるプログラムの例を示す図である。

【図11】図11(a)は、通常のプロセッサが実行する命令のフィールド定義を示す図である。図11(b)は、同命令フォーマットを示す図である。

【図12】図8に示されたプログラムと同一内容の処理を上記通常のプロセッサに行なわせるプログラムの例を示す図である。

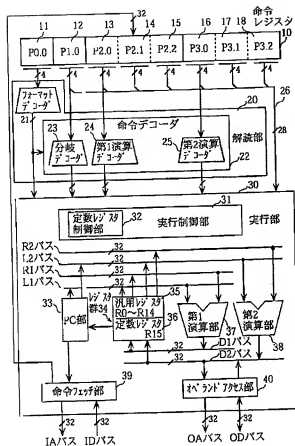
【図13】図10に示されたプログラムと同一内容の処理を上記通常のプロセッサに行なわせるプログラムの例を示す図である。

【図14】図14(a)～図14(d)は、本発明のVLIWプロセッサにかかる命令構造の変形例を示す図である。

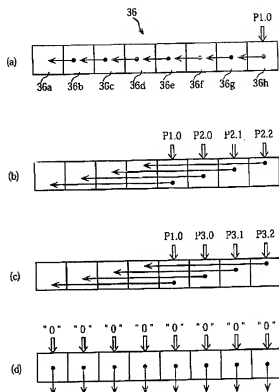
【図15】図15(a)及び図15(b)は、従来技術における命令フォーマットを示し、それぞれ、同時に2個のオペレーションを指定する命令フォーマット、1個



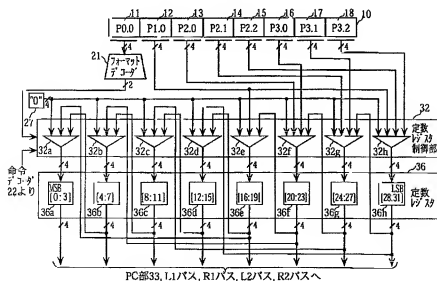
【図3】



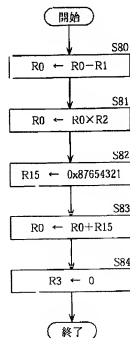
【図5】



【図4】



【図7】





【図8】

P0.0	P1.0	P2.0	P2.1	P2.2	P3.0	P3.1	P3.2
fmt 4	0x8765				Sub	R1	R0
fmt 4	0x4321				mul	R2	R0
fmt 7	0x0	add	R15	R0	mov	0x0	R3

71

72

73

【図9】

クロック  
サイクル

	t0	t1	t2	t3
R0	33333333	11111111	44444444	CBA98765
R1	22222222			
R2		00000004		
R3			FEDCBA98	00000000
R15	00000000	00008765	87654321	00000000
L1			87654321	
R1			44444444	
L2	33333333	11111111		
R2	22222222	00000004	FEDCBA98	

【図10】

P0.0	P1.0	P2.0	P2.1	P2.2	P3.0	P3.1	P3.2
fmt 0	0x8	mov	R6	R1	mov	R7	R2
fmt 0	0x7	add	R0	R1	add	R0	R2
fmt 0	0x6	mul	R6	R1	sub	R7	R2
fmt 0	0x5	mov	R8	R4	mov	R9	R5
fmt 7	0x0	add	R15	R0	mov	0x0	R3

74

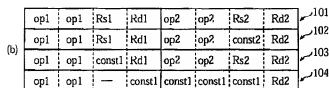
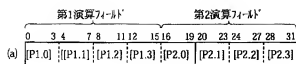
75

76

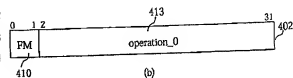
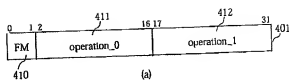
77

78

【図11】



【図15】



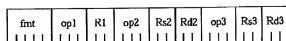
【図12】

P1.0	P1.1	P1.2	P1.3	P2.0	P2.1	P2.2	P2.3
nop	—	—	—	sub	R1	R0	105
nop	—	—	—	mul	R2	R0	106
set hi	—	—	—	0x8/65	—	R15	107
set lo	—	—	—	0x4321	—	R15	108
add	R15	R0	—	mov	0x0	R3	109

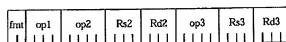
【図13】

P1.0	P1.1	P1.2	P1.3	P2.0	P2.1	P2.2	P2.3
mov	R6	R1	—	mov	R7	R2	110
add	R0	R1	—	add	R0	R2	111
mul	R6	R1	—	sub	R7	R2	112
mov	R8	R4	—	mov	R9	R5	113
add	—	—	—	0x8765	—	R0	114
nop	—	—	—	mov	0x0	R3	115

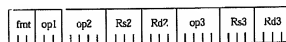
【図14】



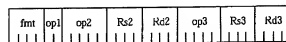
(a)



(b)



(c)



(d)